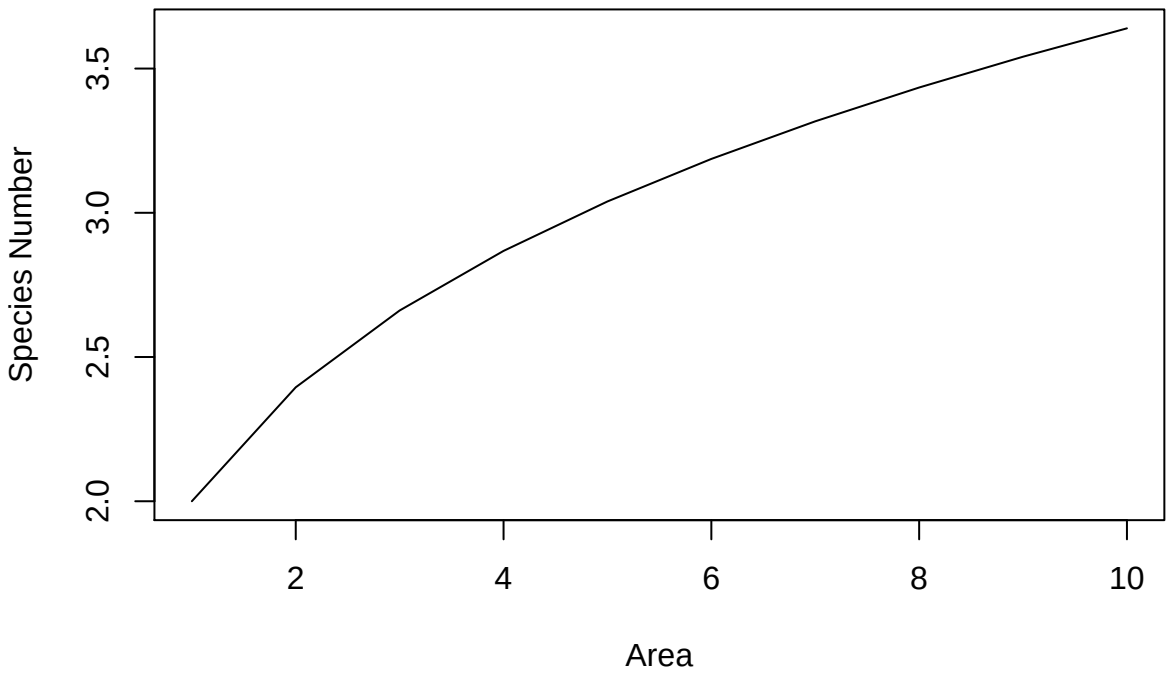
PlottingFunctions.R

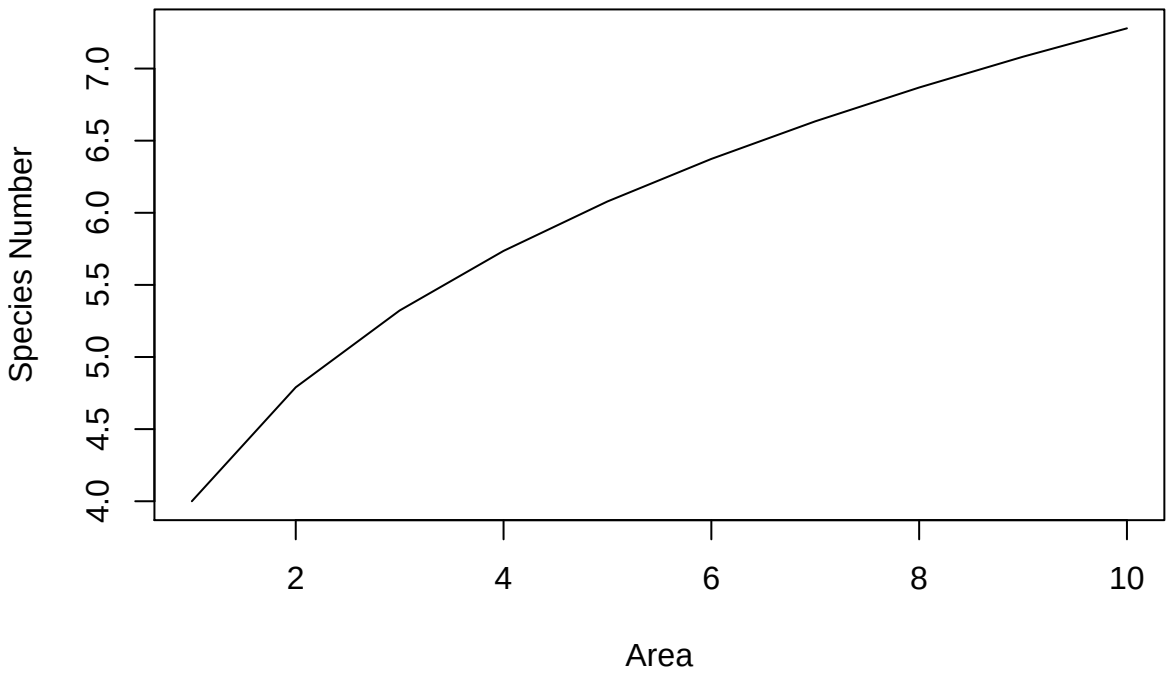
Administrator

Thu Feb 18 14:44:41 2016

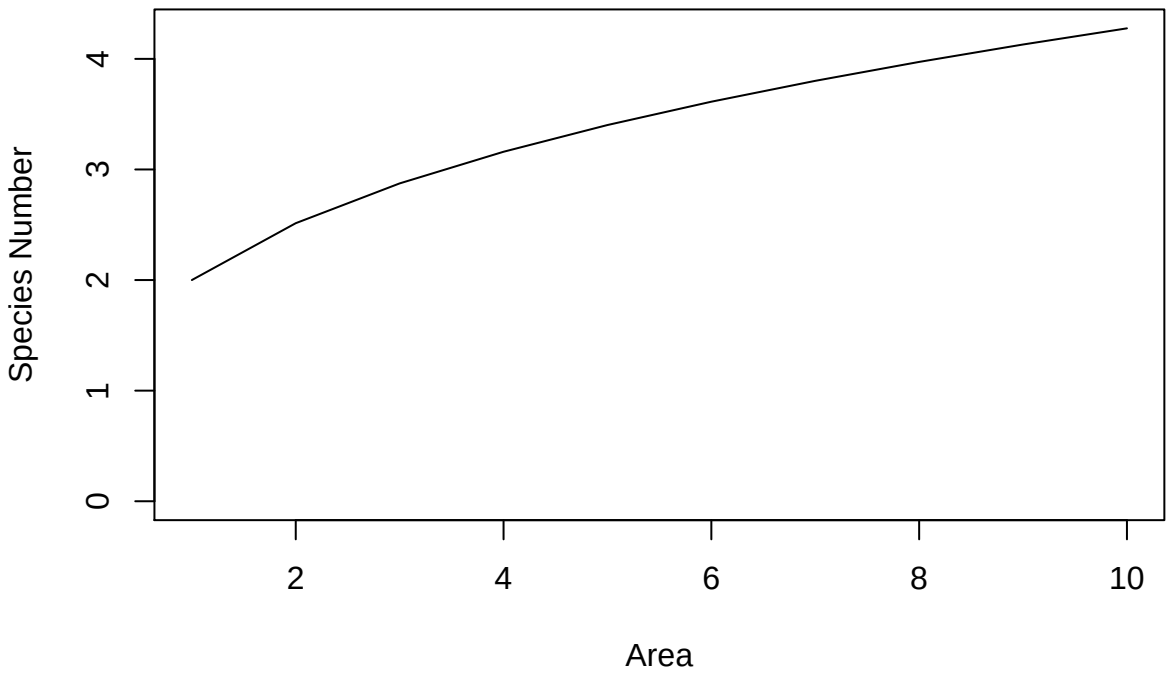
```
# Exploring functions with plots
# 16 Feburary 2016
# NJG

# basic plotting function with inputs
# S = cA^z
# S = number of species (dependent variable)
# A = island or sample area (independent variable)
# c = parameter
# z = parameter

opar <- par()
Plot1 <- function(A=1:10, c=2,z=0.26) {
  S <- c*(A^z)
  plot(x=A,
       y=S,
       xlab="Area",
       ylab="Species Number",
       type="l")
}
Plot1()
```



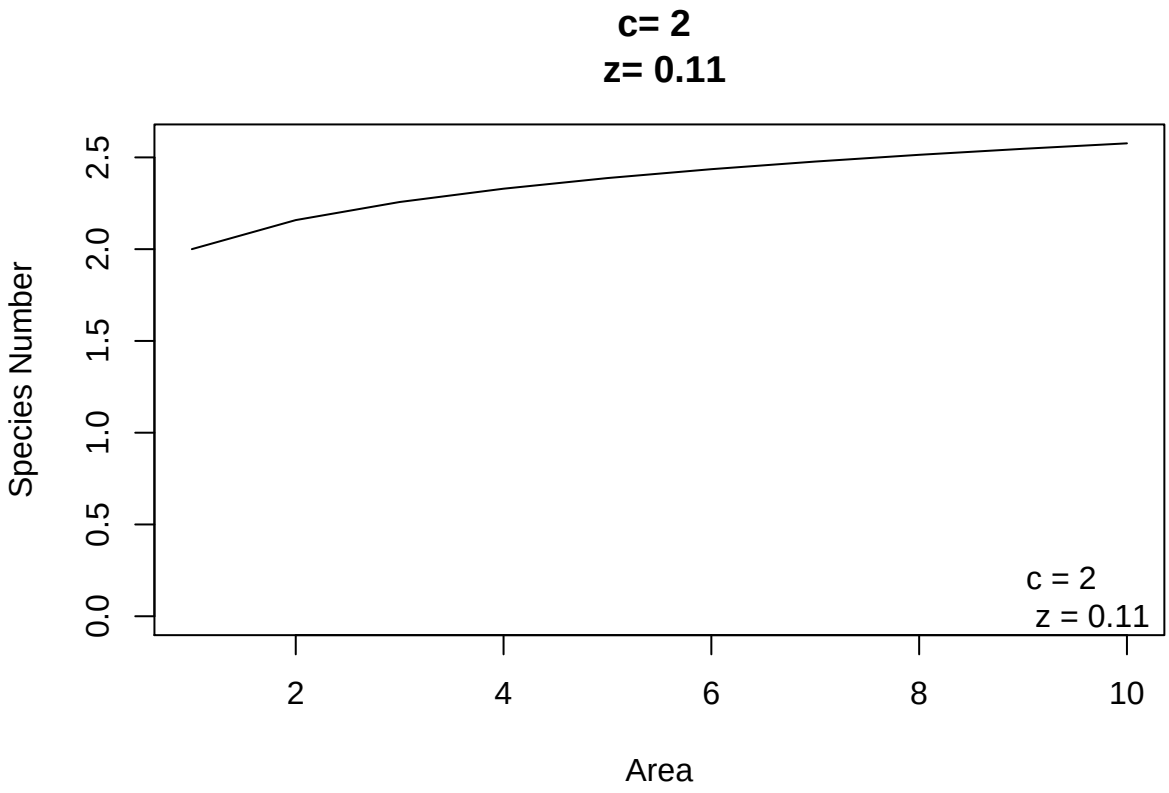

```
# Use different levels of A
# Show linearity with small N
# is there an asymptote
Plot1(c=4)
```



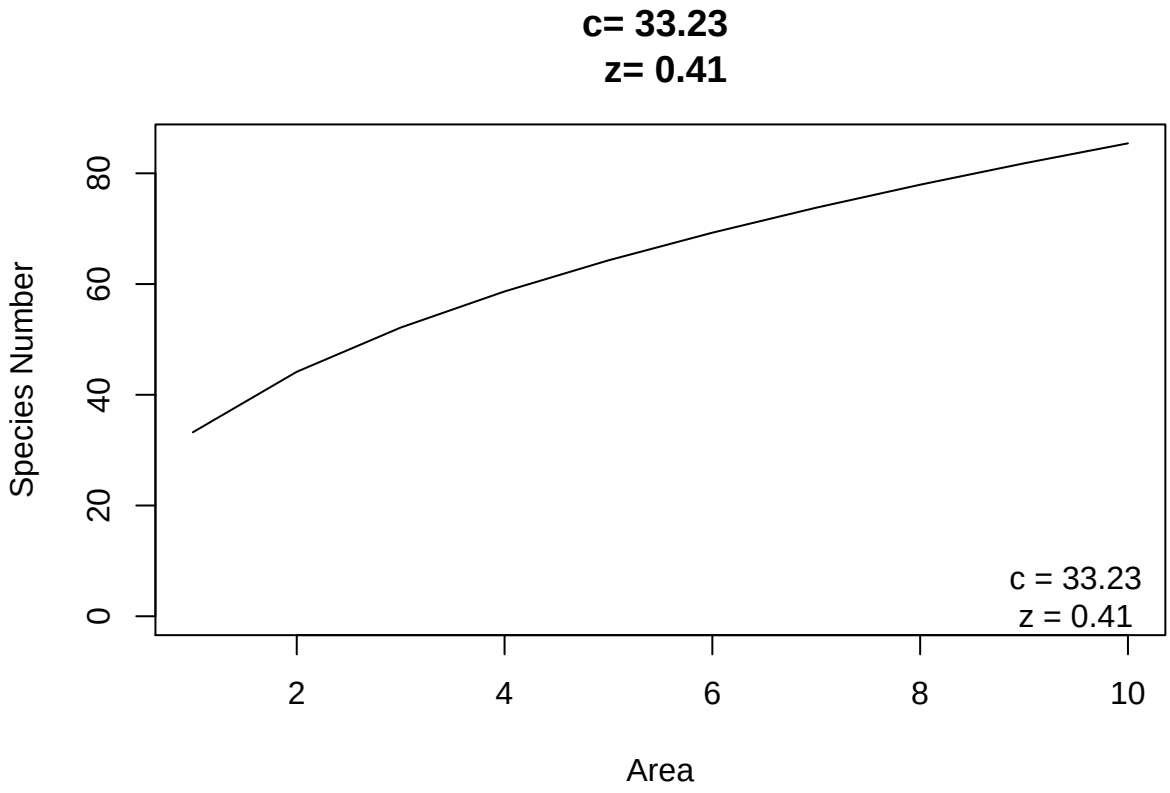

```
# Give more informative limits for graph
Plot2 <- function(A=1:10, c=2,z=0.26) {
  S <- c*(A^z)
  plot(x=A,
       y=S,
       xlab="Area",
       ylab="Species Number",
       type="l",
       ylim=c(0,max(S)))
}
Plot2(z=0.33)
```




```
# Add a legend to show paramter values
# use main or use legend function
Plot3 <- function(A=1:10, c=2,z=0.26) {
  S <- c*(A^z)
  plot(x=A,
       y=S,
       xlab="Area",
       ylab="Species Number",
       type="l",
       ylim=c(0,max(S)),
       main=paste("c=",c,"\n", "z=",z))

  legend(x="bottomright",
         legend= c(paste("c =",c,
                         "\n",
                         "z =",z)),
                bty="n")
}
Plot3(z=0.11)
```

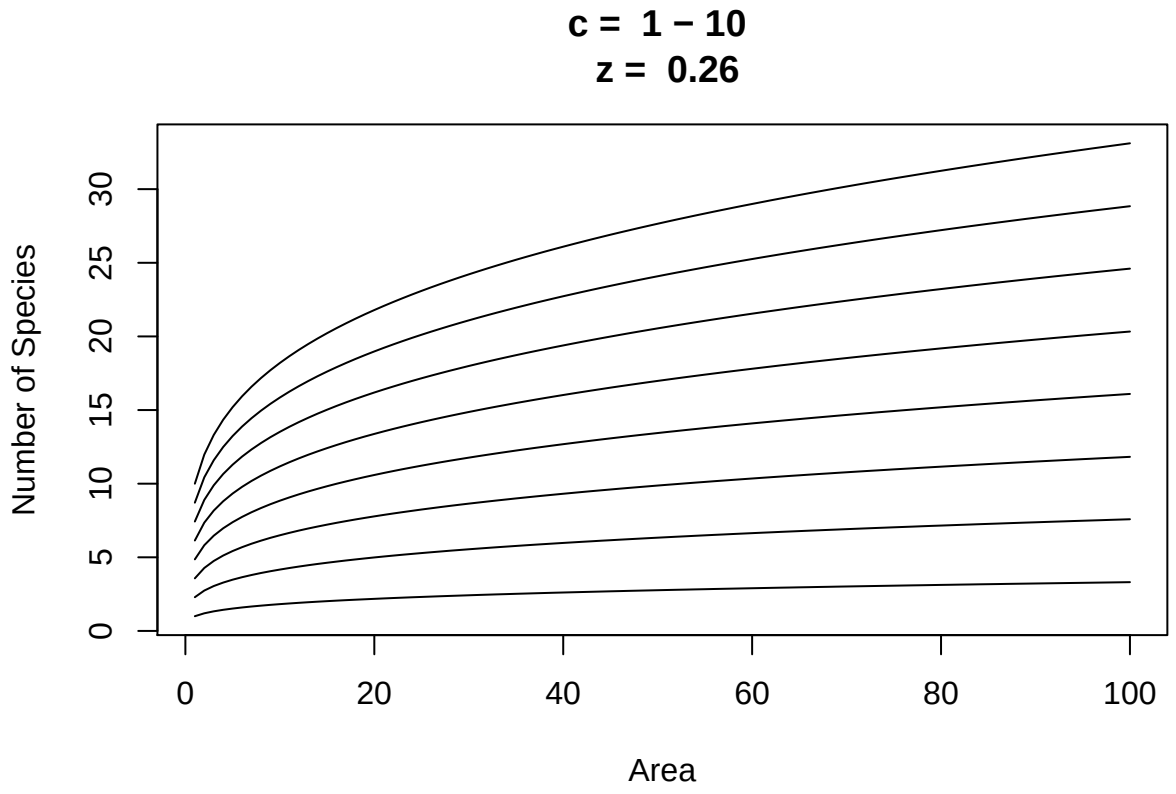



```
# Add random starting values for realistic
# parameter ranges

Plot4 <- function(A=1:10, c=runif(1,min=0,max=50),z=runif(1,min=0,max=1.0)) {
  c <- round(c,2)
  z <- round(z,2)
  S <- c*(A^z)
  plot(x=A,
       y=S,
       xlab="Area",
       ylab="Species Number",
       type="l",
       ylim=c(0,max(S)),
       main=paste("c=",c,"\n", "z=",z))

  legend(x="bottomright",
         legend= c(paste("c =",c,"\n", "z =",z)),
         bty="n")
}
Plot4()
```

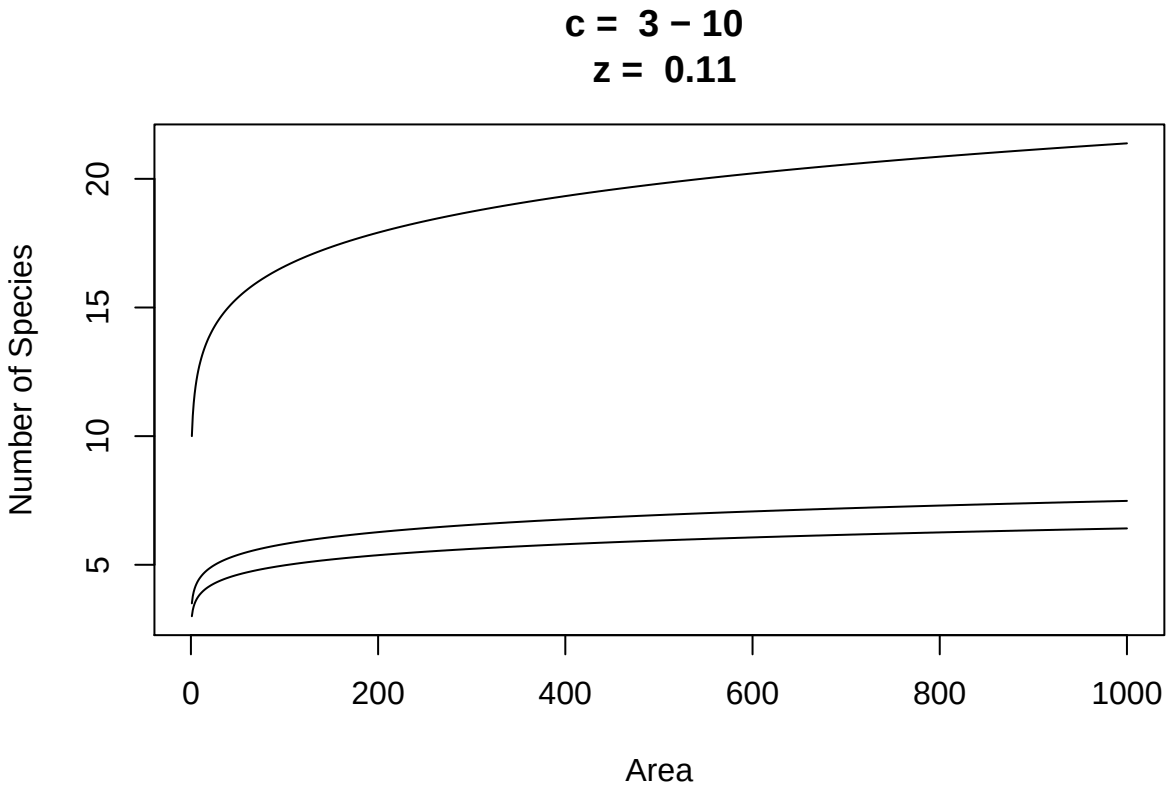



```
# Plot a series of curves

Plot5 <- function(A=1:100, c=seq(1,10,length=8),z=0.26) {
  c <- round(c,2)
  z <- round(z,2)

# Create a matrix to hold the data
# i rows, with one row for each value of A
# j columns, one for each parameter combinatino of z and c
  m <- matrix(nrow=length(A),ncol=length(c))

# Fill the columns of the matrix for the curve
   v <- 1
  for (i in 1:length(c)) {
    S <- c[i]*A^z
    m[,i] <- S
  }
 matplot(x=A,y=m,type="l",
         xlab="Area",
         ylab="Number of Species",
         col="black",
         lty=1,
         # ylim=c(0,30),
         main=paste("c = ",min(c),"-",max(c),
                    "\n",
                    "z = ",z))
```

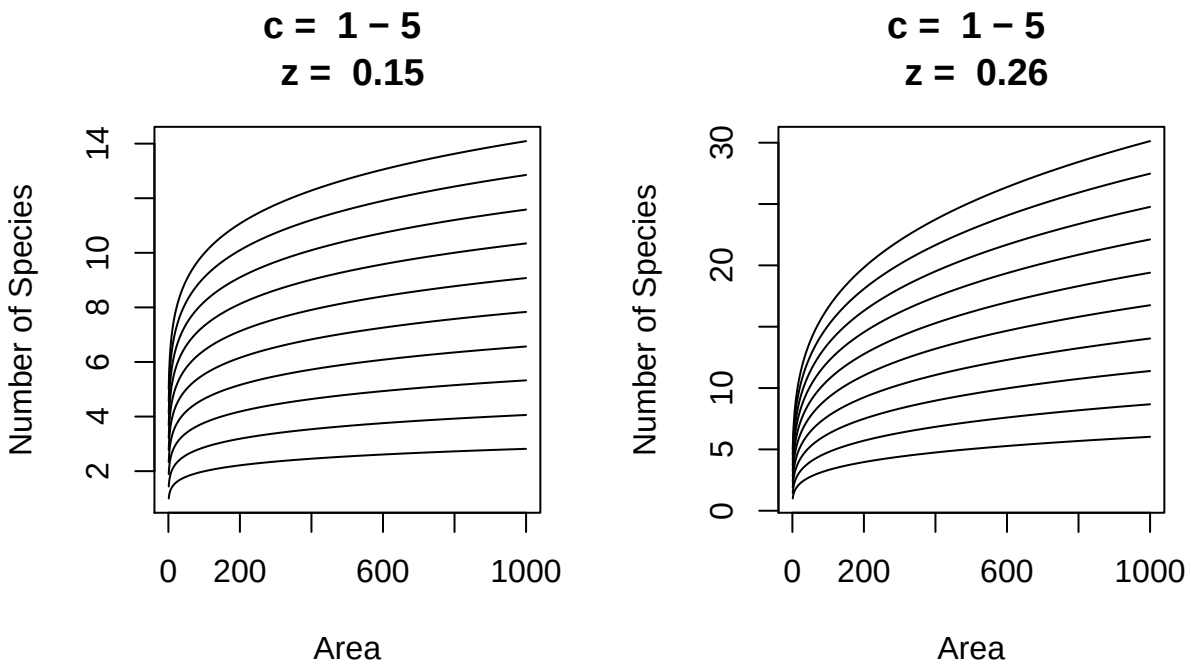
```
}
Plot5()
```




```
Plot5(A=1:1000,c=c(3,3.5,10),z=0.11)
```




```
# Using the function to change more than 1 parameter

par(mfrow=c(1,2),pty="s") # set graphics for 2 plots

myZ <- c(0.15, 0.26) # pick parameters for z

for (x in 1:2) {
  Plot5(A=1:1000, c=seq(1,5,length.out=10),z=myZ[x])
}
```

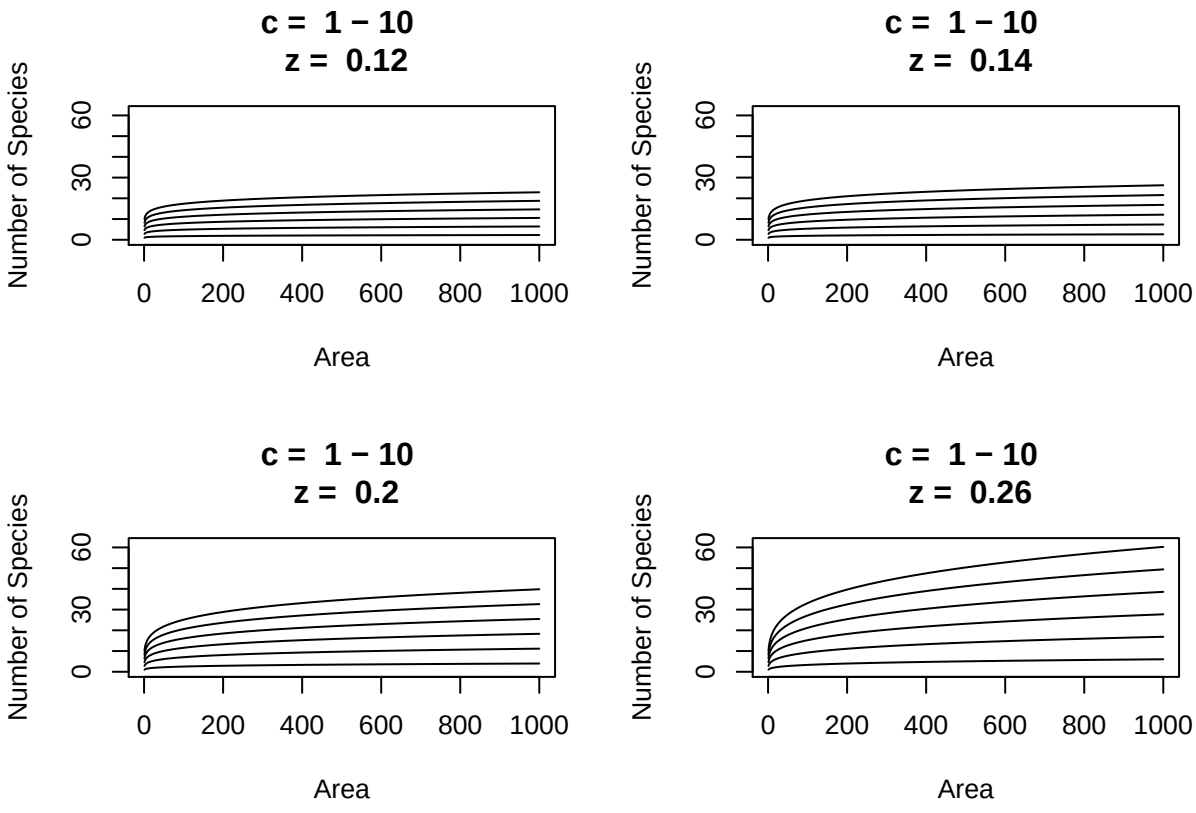



```
par(opar)
```

```
## Warning in par(opar): graphical parameter "cin" cannot be set

## Warning in par(opar): graphical parameter "cra" cannot be set

## Warning in par(opar): graphical parameter "csi" cannot be set

## Warning in par(opar): graphical parameter "cxy" cannot be set

## Warning in par(opar): graphical parameter "din" cannot be set

## Warning in par(opar): graphical parameter "page" cannot be set
```

```
# Add some control over the y axis range
Plot5a <- function(A=1:100, c=seq(1,10,length=8),z=0.26,myY=c(0,100)) {
  c <- round(c,2)
  z <- round(z,2)

# Create a matrix to hold the data
# i rows, with one row for each value of A
# j columns, one for each parameter combinatino of z and c
  m <- matrix(nrow=length(A),ncol=length(c))
```

```
# Fill the columns of the matrix for the curve
   v <- 1
  for (i in 1:length(c)) {
    S <- c[i]*A^z
    m[,i] <- S
  }
 matplot(x=A,y=m,type="l",
         xlab="Area",
         ylab="Number of Species",
         col="black",
         lty=1,
         ylim=myY, # insert the ylimit here
         main=paste("c = ",min(c),"-",max(c),
                    "\n",
                    "z = ",z))
}
par(mfrow=c(2,2))
myZ <- c(0.12, 0.14, 0.20, 0.26)
for (i in 1:length(myZ)){
  Plot5a(A=1:1000,c=seq(1,10,length=6),z=myZ[i],myY=c(0,62))
}
```

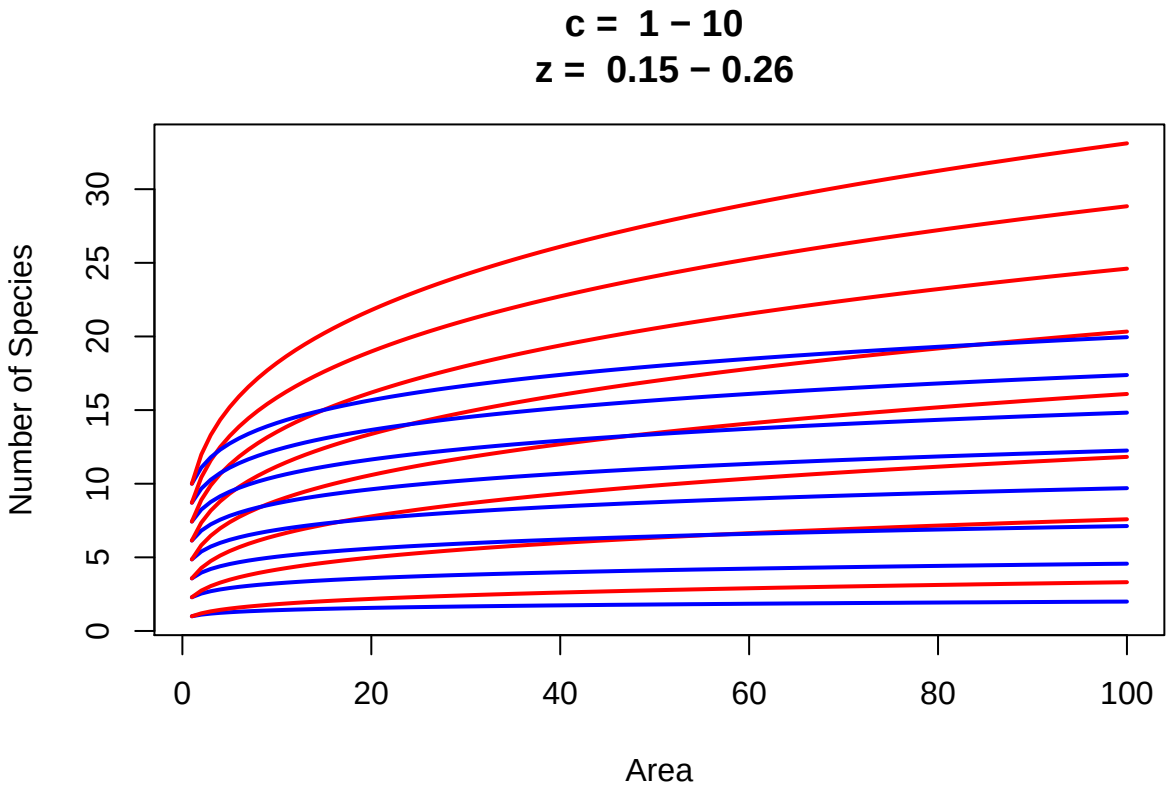



```
par(opar)
```

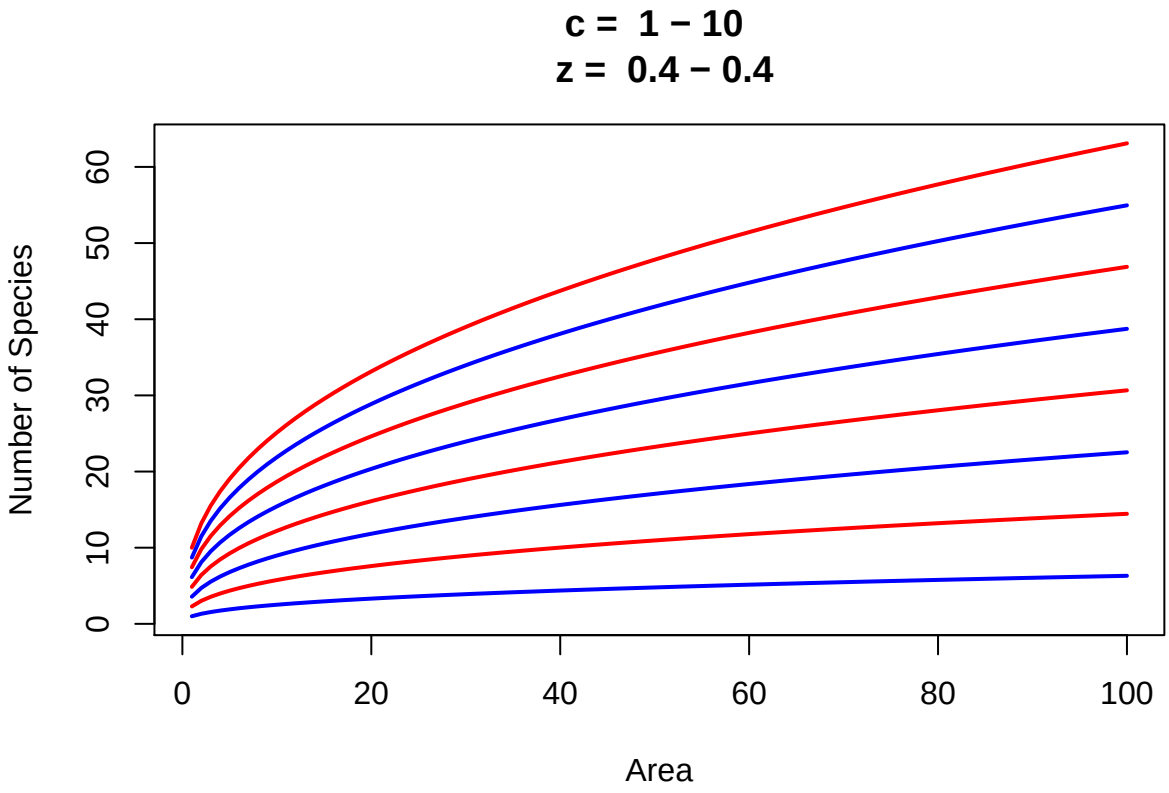
```
## Warning in par(opar): graphical parameter "cin" cannot be set
```

```
## Warning in par(opar): graphical parameter "cra" cannot be set

## Warning in par(opar): graphical parameter "csi" cannot be set

## Warning in par(opar): graphical parameter "cxy" cannot be set

## Warning in par(opar): graphical parameter "din" cannot be set

## Warning in par(opar): graphical parameter "page" cannot be set
```

```
# Redo function to create all parameter combinations first


Plot6 <- function(A=1:100, c=seq(1,10,length=8),z=c(0.15,0.26)) {
  c <- round(c,2)
  z <- round(z,2)

# Create a matrix to hold the data
# i rows, with one row for each value of A
# j columns, one for each parameter combinatino of z and c
  m <- matrix(nrow=length(A),ncol=length(c)*length(z))
  myCols <- rep(c("blue","red"), length(c))

# Fill the columns of the matrix for the curve
   v <- 1
  for (i in 1:length(c)) {
    for (j in 1:length(z)) {
    S <- c[i]*A^z[j]
    m[,v] <- S
    v <- v + 1
    }
  }
 matplot(x=A,y=m,type="l",
         xlab="Area",
         ylab="Number of Species",
         col=myCols,
         lty=1,
         lwd=2,
         main=paste("c = ",min(c),"-",max(c),
                    "\n",
                    "z = ",min(z),"-",max(z)))

}
Plot6()
```


c = 1 − 10
z = 0.15 − 0.26
Number of Species
0
5
10
15
20
25
30
0
20
40
60
80
100
Area

```
Plot6(z=.4)
```


c = 1 − 10
z = 0.4 − 0.4
Number of Species
0
10
20
30
40
50
60
0
20
40
60
80
100
Area